



U.S. DEPARTMENT OF
ENERGY

Office of
Science



MaryLie/IMPACT: Status and Future Plans

Robert D. Ryne

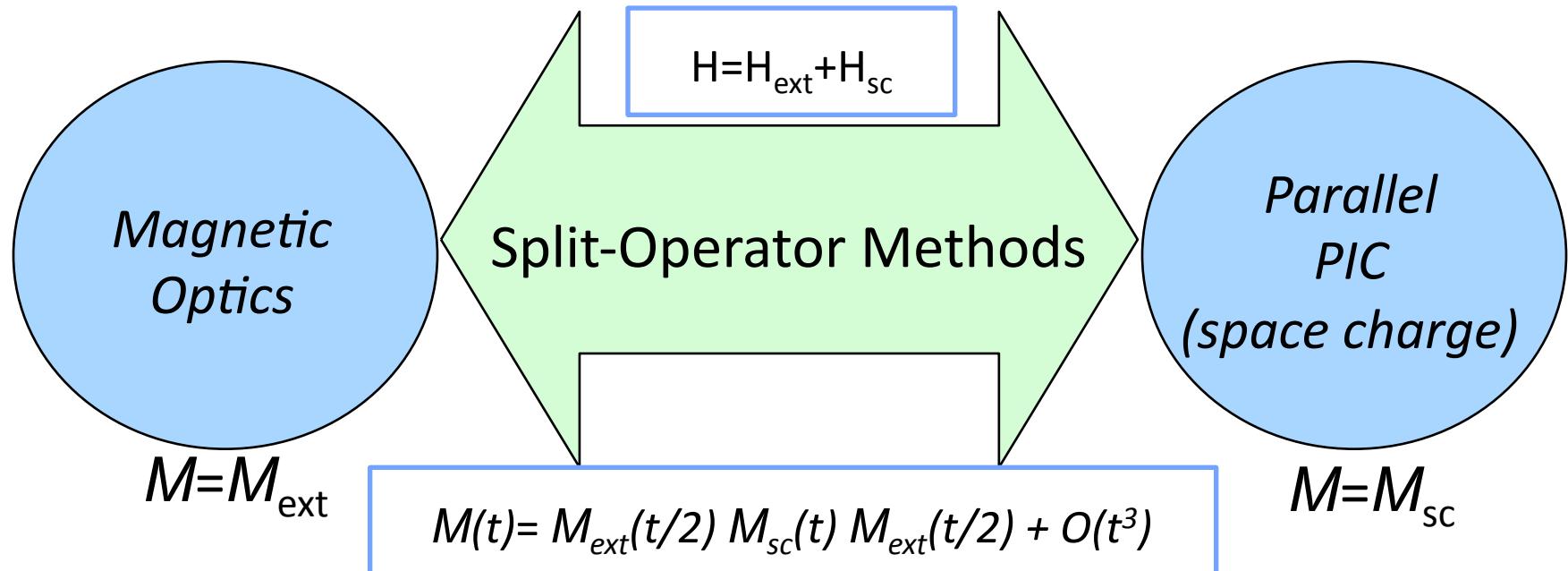
Center of Beam Physics
Lawrence Berkeley National Laboratory

SciDAC II, COMPASS collaboration meeting,
Tech-X Corp. Boulder, CO, Oct. 5-6, 2009

MaryLie/IMPACT: Hybrid Parallel Beam Dynamics Code



- Combines features of MaryLie (U. Md) + IMPACT (LBNL) + new features
- Multiple capabilities in a single unified environment:
 - Map generation, map analysis, particle tracking w/ 3D space charge, envelope tracking, fitting and optimization
 - MAD-style input (Standard Input Format, enhanced as needed)
- Uses split operator method to combine high order optics w/ parallel particle-in-cell



MaryLie/IMPACT: Activities & Improvements



- Ported to ANL/BG systems
- Applied to CERN PS2
 - dynamic apertures studies, space-charge studies, comparisons w/ IMPACT
- Encapsulated D. Abell's 5th order nonlinear map for RF cavity for inclusion in other codes
- New serial scan capability (10 dimensional parameter space)
- New parallel scan capability (10D parameter space)
 - up to 80,000 processors on ANL/Intrepid
- New multi-bunch capability
- Enhanced diagnostics
- Auxilliary scripts to produce animations
- Used in USPAS class (along with Synergia)

Encapsulation of 5th order nonlinear rf cavity model (in collaboration w/ Dan Abell, who wrote the NLRF routines)



- About 20,000 lines of code for everything (map generation, tracking through the map, and all ancillary routines)

```
call get_egengrad(pp1, cpp1)
do jslice=1,nslices
  if(icutslicesinhalf.eq.0)then
    slfrac=1.d0/nslices
    write(6,*)'(nohalf) slfrac=',slfrac
    ihalf=0
    call get_nlrft(pp2, cpp2, h, xmh, jslice, nslices, slfrac, ihalf)
c      <concatenate or else track particles through second half of slice>
    if(idotracking.eq.0)call concat(ah, amh, h, xmh, ah, amh)
    if(idotracking.eq.1)call dotrack(h, xmh, ntaysym, norder, ntrace)
  endif
  if(icutslicesinhalf.eq.1)then
    slfrac=0.5d0/nslices
    write(6,*)'(yeshalf) slfrac=',slfrac
    ihalf=1
    call get_nlrft(pp2, cpp2, h, xmh, jslice, nslices, slfrac, ihalf)
c      <concatenate or else track particles through first half of slice>
    if(idotracking.eq.0)call concat(ah, amh, h, xmh, ah, amh)
    if(idotracking.eq.1)call dotrack(h, xmh, ntaysym, norder, ntrace)
c      <perform space-charge kick>
    ihalf=2
    call get_nlrft(pp2, cpp2, h, xmh, jslice, nslices, slfrac, ihalf)
c      <concatenate or else track particles through second half of slice>
    if(idotracking.eq.0)call concat(ah, amh, h, xmh, ah, amh)
    if(idotracking.eq.1)call dotrack(h, xmh, ntaysym, norder, ntrace)
  endif
enddo
```

Parallel Scan Test Problem: The “GapTest” with scans of the quad strengths



```
#comments
#menu
beam: beam, particle=proton, ekinetic=250.d-3, bfreq=700.d6, bcurr=0.1d0
units: units, type=dynamic, l=1.0d0, w=2.d0*pi*700.d6
multiscan: mdscan, pmin1= 6.0, pmax1= 7.0, np1=25, pname1=fquad[2] &
           pmin2=-6.0, pmax2=-7.0, np2=25, pname2=dquad[2], nprocsppsim=128
dr: drift, l=0.10 slices=4
fquad: quadrupole, l=0.15 g1= 6.00 lfrn=0. tfrn=0. slices=6
dquad: quadrupole, l=0.30 g1=-6.00 lfrn=0. tfrn=0. slices=12
gapa1: rfgap,freq=7.e8,escale=4.e7,phasedeg=45.,file=rfdatal1,steps=100,slices=5
gapb1: rfgap,freq=7.e8,escale=4.e7,phasedeg=-1.,file=rfdatal1,steps=100,slices=5
pois: poisson, nx=64,ny=64,nz=64 !Poisson solver with variable grid size
pois_fixed: poisson, nx=32,ny=32,nz=64 gridsize=fixed, &
             xmin=-.0025, xmax=.0025, ymin=-.0029, ymax=.0029, zmin=-.0095, zmax=.0095
raysin: raytrace, type=readonly file1=partcl.data
slice:autoslice, control=local !this is actually the default
slice_global:autoslice, control=global, l=0.01 !approx 1cm slice thickness
dotrack:autotrack, type=taylor1
post: autoapply, name=prntall
prntmoms: moments, precision=9, nunits=1
prntref: reftraj, precision=9, nunits=1
prntmax: maxsize, precision=9, nunits=1
fileout: pmif, itype=1, isend=3
end: end
cell, line= (fquad dr gapa1 dr dquad dr gapb1 dr fquad)
!prntall, line=(prntmoms,prntref,prntmax)
prntall, line=(prntmoms)
#labor
slice
pois !set poisson solver parameters
raysin !read in some rays
prntall !print moments and reference trajectory
dotrack !tell the code to do autotracking
post !after every slice, print moments
cell !here is the beamline
```

Perform $25 \times 25 = 625$ “point simulations” using 128 processors per point simulation; parcel out the point simulations to each group of 128 procs.

For example, if $NP = 625 \times 128 = 80000$, each group of 128 procs does one point simulation.

If $NP = 40000$, each group of 128 procs does 2 point simulations in succession.

... and so on

Use of methodical coding style helps transition to parallel scan capability



- Store MPI communication variables in a module
 - Everywhere in your code, refer to pre-set, pre-stored variables:
 - lworld (not MPI_COMM_WORLD)
 - idproc (don't compute it “on the fly” in each routine)
- Write to *named files* (never to unnamed fileds, i.e. never to a file that gets a default name like “fort.10”)

If you code in this style, then enhancing your code to do parallel scans is easy. But meeting the prerequisites might be tedious if you have a large code that is not already written in this style of coding!

MaryLie/IMPACT: Next Year Plans



- Prepare tutorial for collaborators on parallel scans w/ MPI groups
- Extend nonlinear rf cavity capability to deflecting modes
- Extend parallel scan capability to prototype parallel optimization capability
- Continued applications to CERN PS2
- Improved user manual and example files